A Preliminary Thesis On"SVC Config Advisor"

KUSHAL S. PATEL

.

Under The Guidance of: Prof. S.A. Shindeassistant Professor, Dept of Comp Enggvpcoe, Baramati. University of Pune
Pune, Maharastra, India

Abstract

The SAN Volume Controller combines software and hardware into a comprehensive, modular appliance that uses symmetric virtualization. In the storage systems, number of heterogeneous hosts is connected to SVC. In SVC it is hectic task to configure all devices and hosts manually. Time requirement to configure elements in SVC set up is very large. Error detection is also difficult as there are number of hosts connected to the system. As there are various hosts connected in the system, it is very much difficult to check the configuration mismatch in the system. Also, if any customer reports an issue, it is hard to and the customer host, switch configuration details and if there is any configuration issue.

This project is introduced to overcome the problems with the configuration of system. This project will be able to get the current configuration of all heterogeneous hosts connected in the system and also able to configure the components. To communicate with remote machines, standard connection establishment mechanisms can be used. Current solution will work on iSCSI-attached hosts. This will reduce required time to configure hosts and to check errors in the configuration. Logs can be maintained to track the internal operations.

Contents

1 Introduction 4
1.1 Introduction 4

2 Literature Survey 7

3 Algorithms and Methods 10
3.1 Config Awareness 10
3.1.1 Storage Stand Elements 10
3.1.2 Workflow / Usecases 13
3.1.3 Internals of System 13

3.2 Con g Suggestion and Auto-Con g mode 18
3.3 Code Maintenance plan 19

4 Conclusion 20

5 References 21

Dedication

Dedicated to
My Friends and Family

Chapter 1: Introduction

1.1 Introduction

A storage area network (SAN) is a dedicated network that provides access to consolidated, block level data storage. SANs are primarily used to make storage devices, such as disk arrays, tape libraries, and optical jukeboxes, accessible to servers so that the devices appear like locally attached devices to the operating system. A SAN typically has its own network of storage devices that are generally not accessible through the local area network by other devices.

The IBM 2145 SAN Volume Controller (SVC) is an inline virtualization or "gateway" device. It logically sits between hosts and storage arrays, presenting itself to hosts as the storage provider (target) and presenting itself to storage arrays as one big host (initiator). SVC is physically attached to any available port in one or several SAN fabrics. The virtualization ap-proach allows for non-disruptive replacements of any part in the storage infrastructure, including the SVC devices themselves. It also aims at simpli-fying compatibility requirements in strongly heterogeneous server and storage landscapes. All advanced functions are therefore implemented in the virtualization layer, which allows switching storage array vendors without impact. Finally, spreading an SVC installation across two or more sites (stretched clustering) enables basic disaster protection paired with continuous avail-ability.

SVC nodes are always clustered, with a minimum of 2 and a current maximum of 8 nodes, and linear scalability. Each node is a 1U high rack-mounted appliance leveraging IBM System x server hardware, protected by redundant power supplies and an integrated 1U high uninterruptible power supply. An integrated two-row display and ve-button keyboard o er stand-alone configuration and monitoring options. Each node has four Fibre Channel ports and two or four 10/1 Gbps Ethernet ports used for FCoE, iSCSI and management. All Fibre Channel and FCoE ports on the SVC are both targets and initiators, and are also utilized for inter-cluster communication. This includes maintaining read/write cache integrity, sharing status information, and forwarding reads and writes.

Storage area network (SAN) configurations that contain SAN Volume Controller nodes must be configured correctly. A SAN configuration that contains SAN Volume Controller nodes must follow configuration rules for the following components:

- Storage systems Nodes
- Fibre Channel host bus adapters (HBAs)
- Converged network adapters (CNAs)
- Fibre Channel switches iSCSI Ethernet ports Fabrics
- Zoning

The SAN Volume Controller combines software and hardware into a comprehensive, modular appliance that uses symmetric virtualization. In the storage systems, number of heterogeneous hosts are connected to SVC. In SVC it is hectic task to configure all devices and hosts manually. Time requirement to configure elements in SVC set up is very large. Error detection is also di cult as there are number of hosts connected to the system. As there are various hosts connected in the system, it is very much di cult to check the configuration mismatch in the system.

This project is introduced to overcome the problems with the con guration of system. This project will be able to get the current configuration of all heterogenous hosts connected in the system and also able to con gure the components. To have communication with remote machines, standard connection establishment mechanisms can be used. Current solution will work on iSCSI-attached hosts. This will reduce required time to con figure hosts and to check errors in the configuration. Logs can be maintained to track the internal operations.

Chapter 2: Literature Survey

A SAN is a specialized, high-speed network attaching servers and storage de-vices. It is sometimes called the network behind the servers. A SAN allows any-to-any connection across the network, using interconnect elements such as routers, gateways, hubs, switches, and directors. It eliminates the traditional dedicated connection between a server and storage, and the concept that the server effectively owns and manages the storage devices. It also eliminates any restriction to the amount of data that a server can access, currently limited by the number of storage devices, which can be attached to the individual server. Instead, a SAN introduces the exibility of networking to enable one server or many heterogeneous servers to share a common stor-age utility, which may comprise many storage devices, including disk, tape, and optical storage. And, the storage utility may be located far from the servers that use it.

The SAN can be viewed as an extension to the storage bus concept, which enables storage devices and servers to be interconnected using similar elements as in local area networks (LANs) and wide area networks (WANs): Routers, hubs, switches, directors, and gateways. A SAN can be shared between servers and/or dedicated to one server. It can be local, or can be extended over geographical distances.

iSCSI:

Internet SCSI (iSCSI) is a transport protocol that carries SCSI com-mands from an initiator to a target. It is a data storage networking protocol that transports 10 Introduction to Storage Area Networks standard Small

Computer System Interface (SCSI) requests over the standard Transmission Control Protocol/Internet Protocol (TCP/IP) networking technology. iSCSI enables the implementation of IP-based storage area networks (SANs), enabling customers to use the same networking technologies from the box level to the Internet for both storage and data networks. As it uses TCP/IP, iSCSI is also well suited to run over almost any physical network. By eliminating the need for a second network technology just for storage, iSCSI will lower the costs of deploying networked storage and increase its potential market.

iSCSI hosts connect to the SAN Volume Controller through the node-port IP address. If the node fails, the address becomes unavailable and the host loses communication with SAN Volume Controller. To allow hosts to maintain access to data, the node-port IP addresses for the failed node are transferred to the partner node in the I/O group. The partner node handles requests for both its own node-port IP addresses and also for node-port IP addresses on the failed node. This process is known as node-port IP failover. In addition to node-port IP addresses, the iSCSI name and iSCSI alias for the failed node are also transferred to the partner node. After the failed node recovers, the node-port IP address and the iSCSI name and alias are returned to the original node.

IBM SAN Volume Controller

The SAN Volume Controller combines software and hardware into a comprehensive, modular appliance that uses symmetric virtualization. Symmet-ric virtualization is achieved by creating a pool of managed disks (MDisks) from the attached storage systems. Those storage systems are then mapped to a set of volumes for use by attached host systems. System administrators can view and access a common pool of storage on the storage area network (SAN). This functionality helps administrators to use storage resources more e ciently and provides a common base for advanced functions.

A SAN is a high-speed Fibre Channel network that connects host systems and storage devices. In a SAN, a host system can be connected to a storage device across the network. The connections are made through units such as routers and switches. The area of the network that contains these units is known as the fabric of the network. SAN Volume Controller software The SAN Volume Controller software performs functions for the host

systems that attach to SAN Volume Controller like: Creates a single pool of storage, Provides logical unit virtualization, Manages logical volumes, Mir-rors logical volumes. The SAN Volume Controller system also provides the following functions like Large scalable cache, Copy Services, IBM FlashCopy (point-in-time copy) function, including thin-provisioned FlashCopy to make multiple targets a ordable, Metro Mirror, (synchronous copy), Global Mirror (asynchronous copy), Data migration,

A system of SAN Volume Controller nodes presents volumes to the hosts. Most of the advanced functions that SAN Volume Controller provides are de ned on volumes. These volumes are created from managed disks (MDisks) that are presented by the RAID storage systems. All data transfer occurs through the SAN Volume Controller nodes, which is described as symmetric virtualization.

The nodes in a system are arranged into pairs known as I/O groups. A single pair is responsible for serving I/O on a given volume. Because a vol-ume is served by two nodes, there is no loss of availability if one node fails or is taken o ine. System management The SAN Volume Controller nodes in a clustered system operate as a single system and present a single point of control for system management and service. System management and error reporting are provided through an Ethernet interface to one of the nodes in the system, which is called the con guration node. The con guration node runs a web server and provides a command-line interface (CLI). The con g-uration node is a role that any node can take. If the current con guration node fails, a new con guration node is selected from the remaining nodes. Each node also provides a command-line interface and web interface for per-forming hardware service actions. I/O operations between hosts and SAN Volume Controller nodes and between SAN Volume Controller nodes and RAID storage systems are performed by using the SCSI standard. The SAN Volume Controller nodes communicate with each other by using private SCSI commands.

Chapter 3: Methods and Internals

3.1 Config Awareness

3.1.1 Storage Stand Elements

In this Phase of project development; con g data is fetched from remote machines, parsed and stored into system variables for future use. There are several elements are attached to the system. Therefore con g awareness is further subdivided into number of small modules depending on element types in the system. The below subsection explains each sub-phases in brief.

1. Host
2. SVC
3. Switches
4. Reachablity and Diagnostic.

1. Host

There are various types of host with di erent Operating Systems are attached to the system . Each type of host may have di erent com-mand sets. This issue can be handled by creating separate class of each type of host. In this project, we are speci cally dealing with following types of hosts:

Host with Linux Operating System

Host with Windows Operating System

Host containing VMWare ESX

(a) Linux Host

In con g awareness of Linux host following data should be collected from Linux machine connected in system.

iSCSI initiator

link detection of all ethx devices

attached multipath status

Installed IO tools

Set MTU Value

(b) Windows Host

To get information form windows host; completely di erent set of In config awareness of Windows host following data should be collected from Linux machine connected in system.

iSCSI initiator

Installed IO tools.

Set MTU Value

System Environmental variables

(c) ESX Host

VMware ESX is an enterprise-level computer virtualization product offered by Vmware. In config awareness phase of ESX host following data is collected :

multipath policy
vswitches information
iSCSI initiator
VNIC information

Host Operating System- all desired host information mention above subsection should be fetched.

2. Platforms

The SAN Volume Controller combines software and hardware into a comprehensive, modular appliance that uses symmetric virtualization. In the storage systems, number of heterogenous hosts are connected to SVC. In our system we need certain info from SVC. In this sub phase of the system; data from SVC is fetched:

portip
Host Objects
Volume info
mdisks attached
mdisk groups
quorums
Cluster health
showfp
lseventlog output

3. Switches

A network switch is a computer networking device that links network segments or network devices. Due to this it is important to track switch if there are any problem in the connection. In this project we are using following data from switch to take further actions:

VLANs on switch
MTU setting of ports
Health check
Extensible support mechanism

4. Reachablity and Diagnosis

Basically all the component in server stand are connected to each other by using interconnection network. Therefore to have communication between components; reachablity is important. In this sub phase of the system; packet reachablity is checked by the system and if there are some error found to reach a particular host then appropriate diag-nostic actions will be taken.

Ping all hosts and nodes attached Check for errors if any

If errors are found then check for solution

3.1.2 Workflow / Usecases

1. User of this tool should be able to install and uninstall this tool without affecting other processes in the system.
2. User should able to add/remove hosts in server rack. Due to this the tool achieves extensibility in terms of hosts.
3. User should able to view / change IP address of hosts in the system.
4. User should able to see con g information of all the hosts attached.
5. Combined con g check can be done by user by selecting multi con g check.
6. User should able to detect any mismatch in con guration and able to get suggestion by tool.
7. Change in configuration of machines connected in the system should be done by tool user if some mismatch is found in configuration.

3.1.3 Internals of System

1. System User Interface

   User interface of this tool is developed using python Cherrypy library module. CherryPy allows to build web applications in much the same way they would build any other object-oriented Python program. This

results in smaller source code developed in less time. HTML code can be coupled with Cherrypy. Due to this more attractive UI can be generated.

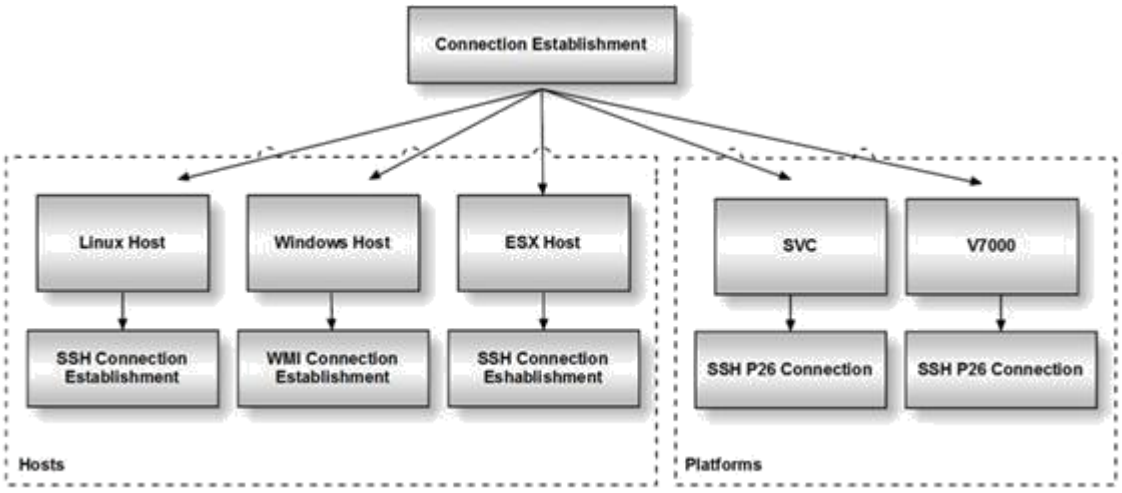
HTML (Hyper-Text Markup Language) is used to generate User inter-face. This can be used in couple with python cherrypy library. This HTML pages are easy to generate and to process. Using Cherrypy; dynamic pages can be created and maintained by the tool.

2. Connection Mechanism

Connection to remote Linux host is established and information is fetched from that host using Secure Shell (SSH) connectivity mech-anism. SSH is a light weight cryptographic network protocol for secure data communication, remote command-line login. Once connection is established, various data can be collected from host.

Connection to remote SAN Volume Controller will be established and information is fetched from SVC. To establish connection, Secure Shell (SSH) connectivity on port 26 can be used. SVC supports SSH connection on port 26. Once connection is established, configuration in-formation will be collected from SAN Volume controller.

In case of windows machine connection, WMI connection mechanism can be used. Windows Management Instrumentation (WMI) is a set of extensions to the Windows Driver Model that provides an operating system interface through which instrumented components provide information and notification. The purpose of WMI is to define a proprietary set of environment independent specifications which allow management information to be shared between management applications. Once WMI connection is established; configuration information will be collected and set from host.



3. Class Structure

To have better connectivity and extensibility of code, a mature class structure should need to be designed. In this project small functions are encapsulated in a def of class. Each class is designed such a way that it should have less coupling factors with other classes in code. Small modules can be created and maintained easily. Modification to the code will be easier if differential modularization is generated for internal and external callable modules. Each class is implemented independently so that it can be given to user for beta testing.

Following classes can created :

(a) SVC-Class : This class contains all the functions related to SVC data fetching. All the defs in this class are used to get data from remote SAN Volume Controller by using its IP address, username and password.

(b) Linux-Class : This class contains all the functions related to Linux host data fetching. This class is used to get data from particular Linux machine in the server rack. Multiple hosts can be handled by creating multiple instances of this class. Minor changes in the command set of Linux hosts can be implemented using overriding of methods of this class.

(c) Class-win : This class contains all the functions related to Windows host data fetching. This class is used to get data from partic-ular Windows machine in the server rack. Multiple windows hosts can be handled by creating multiple instances of this class. Minor changes in the command set due to versioning can be implemented using overriding of methods of this class.

(d) ESX-Data-get : VMware ESX is an enterprise-level computer virtualization product offered by Vmware. This class of ESX data fetching deals with connection establishment with remote ESX host and defs are created to perform data collection task as men-tion above. This class internally may call other class as per virtual operating system installed on this machine.

(e) Home-page : This class contains HTML code for home page of this tool. This class mainly contains data strings of parts of Home page.

(f) Other-Pages : This class contains HTML code for other strings which can be used to display particular sections of web page. Dynamic web page creation can be done by using defs of this class. This class mainly contains defs to get and set particular wedges of web page.

(g) Mismatch-Detect: This class contains implementation of methods to detect mismatch in configuration. This contains defs for mismatch detection for each section. This class takes reference con g information from le and checks for mismatch.

(h) SVC-set-data: This class is used to set con g data on remote SVC node. This will take information from a con g le and set that configuration on remote SAN Volume Controller.

(i) linux-set-data: This class is used to set con g data on remote Linux host. This will take information from a con g le and set that con guration on remote Linux machine.

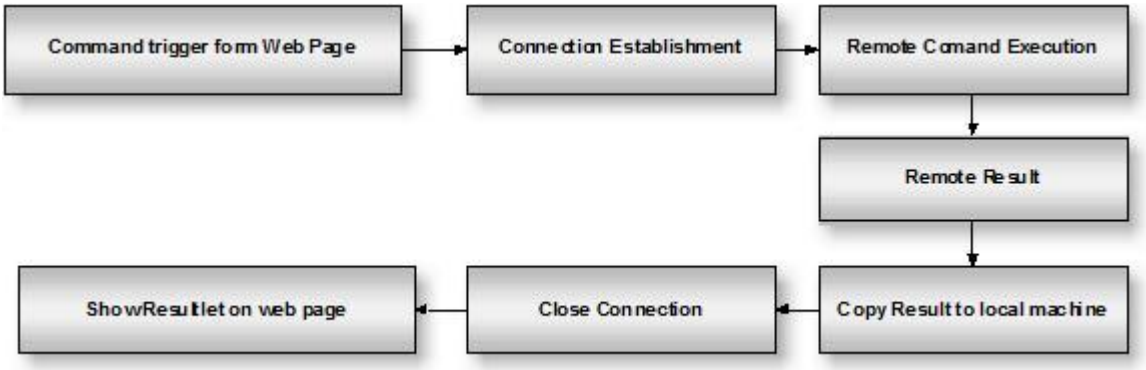
(j) win-set-data : This class is used to set con g data on remote Windows host. This will take information from a con g le and set that configuration on remote Windows machine.

(k) Add-view-elements : This class contains defs to add and remove hosts and nodes. Using this class; tool is able add and remove elements in the server rack.

(l) Tool-Main : This is main class called at the time of running SVC-Con g-Tool.exe. This class contains defs to call other class objects. This class contains defs to control behavior of other classes and to control instances of other classes.

4. Remote Command Execution

To get / set remote data; many remote commands are executed at background. Following steps are done while executing a command on remote machine:



3.1.4 Maintainability and Readability

Software maintenance is an activity that includes error corrections, enhancements of capabilities and optimization. Because change is in-evitable, mechanism must be developed for evaluation, controlling and making modifications. A common perception of maintenance is that it merely involves xing defects. To maintain the code of this project in the future. At the time of generation of source code; we may create the modules which are small in size. Due to the small size of module; functional working will be easier to understand. Small modules can be created and maintained easily. Modification to the code will be easier if differential modularization is generated for internal and external callable modules. There can be Adaptive maintenance in which modification in the system to cope with changes in the software environment

is done. Perfective code maintenance plan implements new or changed user requirements which concern functional enhancements to the soft-ware whereas Corrective maintenance plan deals with diagnosing and xing errors, possibly ones found by users. Preventive plans can be used to Increase software maintainability or reliability to prevent problems in the future.

In this project classes are created in such a way that coupling between class instances are minimal with high cohesion. Due to this independent classes are created and can be maintain in simpler way. Each class contain a complete speci cations about that part of project.

3.2 Config Suggestion and Auto-Config mode

This phase deals with the necessary data collected from all the heterogeneous nodes connected in the system. In this phase, con guration check process is done and error in configuration will be detected. If any mismatch found then user notification can be raised. Log of notifications will be maintain at master host. In this case preferred con guration settings are suggested to user. Configuration of outliers may be corrected automatically if auto correction mode is on. In this phase same connection mechanisms can be used to connect remote entities.

At this stage system should able to detect and correct mismatch in configuration of nodes attached in the system.

GUI is an integrated stage with this phase. This is nal stage of tool. At this stage user interface will be created. Designing the visual composition and temporal behavior of GUI is an important part of software application programming in the area of human-computer interaction. To create UI, wx-Python tool can be used. wxPython is an open-source Python interface for Windows operating system. Handling any software by command line in-terface is efficient but error-prone when the user needs to enter very long commands. User interface allows users to interact with electronic devices through graphical icons and visual indicators. WxPython provides standard API to create UI in python.

3.3 Code Maintenance plan

Software maintenance is an activity that includes error corrections, enhancements of capabilities and optimization. Because change is inevitable, mechanism must be developed for evaluation, controlling and making modi cations. A common perception of maintenance is that it merely involves xing defects. To maintain the code of this project in the future. At the time of generation of source code; we may create the modules which are small in size. Due to the small size of module; functional working will be easier to understand. Small modules can be created and maintained easily. Modification to the code will be easier if differential modularization is generated for internal and external callable modules. There can be Adaptive maintenance in which modification in the system to cope with changes in the software environment is done. Per-fective code maintenance plan implements new or changed user requirements which concern functional enhancements to the software whereas Corrective maintenance plan deals with diagnosing and xing errors, possibly ones found by users. Preventive plans can be used to Increase software maintainability or reliability to prevent problems in the future.

Chapter 4: Conclusion

This tool deals with number of heterogeneous network components and will provide an integrated platform for all these networked devices. This will increase the efficiency to handle the networked component of Storage Area Network and to manage IBM SAN Volume Controller Stand. Typically a storage stand management issue will get resolved with the implementation of this tool. Also time required detecting any misbehavior and misconfiguration of storage stand becomes easier. Complete health checking and individual health checking reduces time required to detect error in configuration of stand. Integrated Evidence-Logger-Tool can be used by tester to capture the logs after every exit of test trial. This will stores all the logs to a specified server with a login and password credentials.

References

1. Whitepaper References

(a) "IBM System Storage"SAN Volume Controller 7.1 Entry Edition and DS3400 36,000 Mailbox Resiliency Exchange 2010 Storage Solution" IBM pvt. Ltd., May 2012

(b) Nick Clayton and Carlos F Fuente, "Planning for Easy Tier with IBM System Storage" Storwize V7000 and SAN Volume Con-troller," Document ID: WP102295, May 2008

(c) Ramapriya Krishnamurthy, "IBM SAN Volume Controller Perfor-mance Con guration Guidelines for Implementing Oracle Databases with Automatic Storage Management (ASM)," Document ID: WP101481

(d) Stephen Wehr, "High Availability Architectures For Linux on IBM System z,"Document ID: WP100752, Augest 2012

(e) Wilhelm Gardt, "Best Practices for Tivoli Storage FlashCopy Manager in a SVC Metro Mirror environment," Document ID: WP102196, June 2011

(f) Narayana Pattipati and Frank Battaglia,"SAS deployment on IBM Power servers with IBM PowerVM dedicated-donating LPARs," Document ID: WP102244, April 2013

2. Web References

(a) http://www-03.ibm.com/systems/storage/software/virtualization/svc/
(b) http://pic.dhe.ibm.com/infocenter/svc/ic/index.jsp
(c) http://www.ietf.org/rfc/rfc1813.txt
(d) http://www.ietf.org/rfc/rfc3530.txt
(e) http://www-03.ibm.com/support/techdocs/atsmastr.nsf/Web/WPByProduct
(f) http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101613